ISSN 1561-5359. Штучний інтелект, 2018, № 2

УДК 519.6

А.Н. Нестеренко, Т.О. Герасимова, І.А. Баранов
Інститут кібернетики імені В.М. Глушкова НАН України, Україна
пр. Академіка Глушкова, 40, м. Київ, 03187

ДЕЯКІ ПІДХОДИ ДО РОЗРОБКИ ПАРАЛЕЛЬНИХ АЛГОРИТМІВ РОЗВ'ЯЗУВАННЯ ЗАДАЧ НА КОМП'ЮТЕРАХ З ПРОЦЕСОРАМИ INTEL XEON PHI

A.N. Nesterenko, T.O. Gerasimova, I.A. Baranov
V.M. Hlushkov Institute of Cybernetics, NAS of Ukraine, Ukraine
40, Academician Hlushkov av., Kyiv, 03187

SOME APPROACHES TO DEVELOP PARALLEL ALGORITHMS FOR SOLVING TASKS ON COMPUTERS WITH INTEL XEON PHI PROCESSORS

В роботі розглядаються алгоритми методу Ньютона для розв'язування систем нелінійних рівнянь (СНР) та методу Рунге-Кутта четвертого порядку для розв'язування задач Коші для систем звичайних диференціальних рівнянь (СЗДР) для багатоядерних комп'ютерів з процесорами Intel Xeon Phi. При розробці алгоритмів використовувалась багаторівнева модель паралельних обчислень та враховувалися особливості архітектури багатоядерного комп'ютера. Наведено часи розв'язування СНУ і СЗДР різних порядків, обраховані коефіцієнти прискорення і ефективності використання запропонованих методів.

Ключові слова: багатоядерні комп'ютери, системи нелінійних рівнянь, задачі Коші для систем звичайних диференціальних рівнянь

The paper deals with the algorithm of the Newton method for solving nonlinear systems (NLS) and the fourth-order Runge-Kutta method for solving Cauchy problems for systems of ordinary differential equations (SODE) on multi-core computers with Intel Xeon Phi processors. In the development of algorithms, a multi-level model of parallel computing was used and features of the architecture of the multi-core computer were taken into account. Times required for the solving of various order SNE and SODE are given; acceleration and performance coefficients characterizing the employment of methods being proposed are evaluated, as well.

Keywords: multi-core computers, non-linear systems, initial-value problems for systems of ordinary differential equations

Вступ

Одним із основних засобів вивчення процесів і явищ різної природи, що виникають у суспільстві, економіці, науці та техніці, є чисельне моделювання. При чисельному моделюванні в різних галузях науки і техніки дуже часто виникають розрахункові задачі, які є або системами нелінійних рівнянь (СНР), або задачами з початковими умовами для систем звичайних диференціальних рівнянь (СЗДР).

При розв'язуванні деяких задач, наприклад, пов'язаних із рухом керованих об'єктів, виникає необхідність розв'язувати СЗДР або СНР високого порядку швидше, ніж відбувається процес у реальному часі; більш того, їх розв'язування потребує багатоваріантних розрахунків та значних обчислювальних ресурсів.

Розрахунок таких задач на комп'ютерах вимагає відповідного збільшення продуктивності комп'ютерів. Збільшення продуктивності комп'ютерів тісно пов'язано зі збільшенням кількості процесорів, що входять до складу паралельного комп'ютера. Але збільшення числа процесорів у паралельних комп'ютерах часто призводить до значного збільшення комунікаційних втрат і зниження ефективності їх використання.

Вимоги до високопродуктивної обчислювальної техніки випереджають можливості традиційних паралельних комп'ютерів. Тому особливу увагу приділяють розробці суперкомп'ютерів принципово нової архітектури. До таких суперкомп'ютерів належать суперкомп'ютери, побудовані на розроблених корпорацією

Intel багатоядерних процесорах Intel Xeon Phi [1]. Нове покоління процесорів Intel Xeon Phi x200 є процесорами-прискорювачами. Вони можуть повністю замінити центральні процесори у паралельному комп’ютері [2]. Це, зокрема, означає можливість виконання без перекомпіляції всіх наявних програм і зменшення складності, пов'язаної із забезпеченням одночасного використання центральних і графічних процесорів в одній системі.

Кожний з таких прискорювачів побудований на мікросхемах, що містять до 36 процесорних «плиток» (tile). Кожна «плитка» містить два процесорних ядра з двома VPU (векторними процесорними пристроями) для роботи з числами в форматі з плаваючою комою подвійної точності. Прискорювач може працювати як з високопродуктивною вбудованою пам’яттю MCDRAM (Multi-Chennel DRAM), так і з оперативною пам’яттю DDR4, що дозволяє обробляти великі обсяги даних.

Разом зі зростанням можливостей комп’ютерів змінюються і підходи до створення паралельних алгоритмів. Якщо раніше необхідно було створювати паралельні алгоритми розв’язування задач на однотипних процесорах, то зараз при розробці паралельних алгоритмів необхідно враховувати різні архітектури обчислю0вальних ресурсів, щоб алгоритми були ефективними і давали можливість якнайшвидше отримати розв’язок задач та мінімізувати комунікаційні витрати.

Зупинимось на питанні розробки алгоритмів і програм та організації обчислень при розв’язуванні задач з початковими умовами для СЗДР та СНР на паралельних комп’ютерах з багатоядерними процесорами Intel Xeon Phi x200.

Структура процесорів Intel Xeon Phi дає можливість використання кількох рівнів паралелізму:

- паралелізм рівня обчислювальних вузлів (process level parallelism - PLP) дає можливість проводити обчислення у вигляді паралельних процесів, передбачає використання розподіленої між паралельними процесами пам’яті, забезпечує синхронізацію обчислень та обмін інформацією між процесами;
- паралелізм рівня потоків (thread level parallelism – TLP) передбачає використання потоків, які працюють із загальною пам'яттю;
- паралелізм обробки даних векторними арифметико-логічними пристроями (data level parallelism - DLP) дає можливість автоматичного включення паралелізму в програму за допомогою компілятора.

З метою отримання ефективних алгоритмів розпаралелювання необхідно:

- розподілити задачу на підзадачі таким чином, щоб мінімізувати кількість і обсяг інформаційної залежності між ними;
- визначити рівні паралелізму та середовище розпаралелення;
- врахувати структуру пам’яті процесорних пристроїв;
- розподілити дані та обчислення між процесами, забезпечивши рівномірне завантаження обчислювальних вузлів комп’ютера.

Постановка задач з наближеними вихідними даними для систем нелінійних рівнянь

Нехай дана система n нелінійних рівнянь

$$f(x)=0, \quad (1)$$

де $f(x)=\left(f_1(x), f_2(x), \ldots, f(x)\right)^T$ – n-вимірна вектор-функція, а $x=\left(x_1, x_2, \ldots, x_n\right)^T$ – n-вимірний вектор, причому $f(x)=0$ є деяким наближенням до точної системи нелінійних рівнянь $\varphi(y)=0$, і для цих вектор-функцій виконується нерівність

$$\left\| f(u)-\varphi(u) \right\| \le \delta \quad (2)$$

на будь-якому n-вимірному векторі u. Для розв’язування задачі (1) задаються: початкове наближення $x^{(0)}$, область $D=\left\{a_i \le x_i \le b_i, \quad i=1,2,\ldots,n\right\}$, в якій шукається розв’язок, необхідна точність ε отримання наближення до розв’язку системи. При цьому $x^{(0)} \in D$. Нижнім індексом у формулах позначені номери

компонент векторів, а верхнім – номери ітерацій.

Якщо $H = \left\{ \frac{\partial f_i}{\partial x_j} \right\}_{i,j=1}^{n}$ – матриця Якобі системи (1) (або деяке наближення до неї), то ітераційний процес методу Ньютона знаходження розв'язку при заданому початковому наближенні можна записати у вигляді

$$H^{(k)} w^{(k)} = -f\left(x^{(k)}\right), \qquad (3)$$

де $w^{(k)} = x^{(k+1)} - x^{(k)}$ – поправка, $k = 0,1$ ... – номер ітерації, і

$$x^{(k+1)} = x^{(k)} + w^{(k)}. \qquad (4)$$

Як видно з формули (3), на кожній ітерації необхідно розв'язувати систему лінійних алгебраїчних рівнянь, обчислюючи значення вектор-функції і матрицю Якобі.

Для отримання розв'язку системи нелінійних рівнянь (1) з заданою точністю $\left\| x^{(k)} - x \right\| \le \varepsilon$ ітераційний процес необхідно закінчувати при виконанні умови

$$\left\| f\left(x^{(k)}\right) \right\| \le \frac{\varepsilon}{\left\| H^{-1^{(k)}} \right\|}, \qquad (5)$$

де $H^{-1^{(k)}}$ – матриця, обернена по відношенню до матриці Якобі, обчисленої на k-ій ітерації [3]. Але оскільки для перевірки умови (5) на кожній ітерації необхідно обертати матрицю Якобі, що вимагає значної кількості арифметичних операцій, то пропонується на перших ітераціях скористатися більш економною умовою закінчення ітерацій $\left\| f\left(x^{(k)}\right) \right\| \le \varepsilon$, а після її виконання переходити до перевірки умови (5).

Отже, задачу знаходження розв'язку СНР можна розділити на наступні підзадачі:

1) обчислення вектор-функції;
2) обчислення матриці Якобі;
3) розв'язування отриманої системи лінійних алгебраїчних рівнянь (СЛАР) за формулою (3);
4) обчислення наступного наближення до розв'язку СНР за формулою (4), використовуючи отриманий розв'язок СЛАР;
5) перевірка умов закінчення ітерацій-ного процесу;
6) оцінка якості наближеного розв'язку за формулою (5).

Паралельний алгоритм методу Ньютона

На рівні PLP паралелізму обчислювальних вузлів паралельний алгоритм методу Ньютона реалізується в середовищі MPI з розподіленою пам'яттю на *p* процесах. Тому до початку ітераційного процесу в пам'ять кожного з *p* MPI процесів посилається вхідна інформація (порядок системи, вектор початкового наближення до розв'язку, межі області, в якій шукається розв'язок, похибка в початкових даних, необхідна точність отримання розв'язку та ін.). З метою забезпечення рівномірного завантаження обчислювальних елементів комп'ютера виконується автоматичний розподіл обчислення значень компонент *n*-вимірної вектор-функції на *p* блоків (*p* – кількість MPI процесів) [3, 4].

Підзадачі 1) – 5) виконуються на кожній ітерації алгоритму методу Ньютона, а підзадача 6) – по завершенню ітераційного процесу у вигляді паралельних MPI процесів.

Обмін даними між процесами для підзадач 3), 4), 6) відбувається за стандартами MPI (Message Passing Interface – Інтерфейс Передачі Повідомлень), який описує способи обміну повідомленнями між паралельними процесами в системах з розподіленою пам'яттю.

Оскільки для кожного з MPI процесів підзадачі 1) – 6) – виконуються однорідні операції над великим обсягом даних, то наступний крок розпаралелювання доцільно реалізовувати в середовищі OpenMP (Open Multi-Processing) – сукупність директив компілятора, бібліотечних функцій і змінних оточення, призначених для програмування багатопотокових додатків у багатоядерних і багатопроцесорних систе-

мах із загальною пам'яттю. Це другий рівень паралелізму – рівень паралелізму потоків. Тут кожна з підзадач 1) – 6) верхнього рівня паралелізму розпаралелюється між деякою кількістю потоків (threads) на вільних ядрах з використанням спільної пам’яті.

На третьому рівні паралелізму – паралелізм обробки даних – проводиться автоматична векторізація циклів за рахунок використання векторних процесорних пристроїв (VPU).

Постановка задачі Коші для системи звичайних диференціальних рівнянь

Нехай дана система n звичайних диференціальних рівнянь. Задачу з початковими умовами (задачу Коші) для СЗДР n-го порядку на інтервалі $[t_0, T]$ розглядатимемо у вигляді:

$$\frac{dy}{dt} = f(t, y), \tag{6}$$

$$y(t_0) = y_0 , \tag{7}$$

де $y = (y_1, y_2, \ldots, y_n)^T$ – шуканий вектор, а права частина системи – n-вимірна неперервна вектор-функція

$$f(t, y) = (f_1(t, y), f_2(t, y), \ldots, f_n(t, y))^T .$$

При моделюванні реальних процесів на комп'ютері за допомогою СЗДР виникає ряд труднощів, зокрема, потреба мати справу із задачами з наближеними вихідними даними. Наближений характер вихідних даних може бути обумовлений наступними причинами:

- похибками у вихідних даних;
- похибками у значеннях правої частини;
- застосуванням чисельного (дискет-ного) методу інтегрування і округленням чисел при обчисленнях;
- дискретизацією динамічних задач по просторових змінних.

Тому на практиці, як правило, замість задачі (6), (7) маємо задачу з наближеними вихідними даними:

$$\frac{dv}{dt} = f(t, v), \tag{8}$$

$$v(t_0) = v_{0,} \tag{9}$$

де $\|y_0 - v_0\| \le \delta$, $\varphi(t, w) = f(t, w) + \Delta(t, w)$, $\|\Delta(t, w)\| \le \Delta$ для довільних функцій $w(t)$.

Такі задачі і розв’язуються на комп'ютерах за допомогою чисельних методів. Одним із методів, що застосовується для чисельного інтегрування задач з початковими умовами для СЗДР, є метод Рунге-Кутта 4-го порядку. Цей метод є найбільш розповсюдженим методом чисельного розв’язування задач з початковими умовами для СЗДР.

Класичний метод Рунге–Кутта 4-го порядку реалізується за формулами:

$$y^{(i+1)} = y^{(i)} + (k_1 + 2k_2 + 2k_3 + k_4)/6, \tag{10}$$

де

$$\begin{aligned} k_1 &= h_i f(t_i, y^{(i)}), \\ k_2 &= h_i f(t_i + h_i/2, y^{(i)} + 0{,}5k_1), \\ k_3 &= h_i f(t_i + h_i/2, y^{(i)} + 0{,}5k_2), \\ k_4 &= h_i f(t_i + h_i, y^{(i)} + k_3), \end{aligned} \tag{11}$$

i=0,1,2,… верхній індекс – номер точки, нижній індекс – номер компоненти вектора.

Таким чином процес знаходження розв’язку задачі з початковими умовами для СЗДР методом Рунге-Кутта 4-го порядку можна розділити на наступні підзадачі, які виконуються як паралельні MPI процеси на p ядрах (паралелізм рівня обчислювальних вузлів):

1) обчислення вектор-функції $f(t, y)$;
2) обчислення векторів k_1, k_2, k_3 та k_4 за формулами (11) ;
3) обчислення вектора розв’язку $y = (y_1, y_2, \ldots, y_n)^T$ у наступній точці інтегрування, виходячи із уже обчисленого розв’язку у попередній точці (попередніх точках) за формулою (10);
4) двократним інтегруванням з довжиною кроку $h_i/2$ обчислюється вектор $\bar{\bar{y}}^{(i+1)}$ та однократним інтегруванням з довжиною кроку h_i обчислюється вектор $\bar{y}^{(i+1)}$. Після чого обчислюється похибка апроксимації системи за формулою $\psi^{(i+1)} = \max_{1 \le j \le n} \left| \bar{y}_j^{(i+1)} - \bar{\bar{y}}_j^{(i+1)} \right|$;

5) перевірка умови досягнення заданої точності;
6) перевірка умови досягнення кінцевої точки інтервалу інтегрування;
7) обчислення константи Ліпшиця.

Паралельний алгоритм методу Рунге-Кутта 4-го порядку точності

При розв'язуванні задач з початковими умовами для СЗДР значна частина арифметичних операцій припадає на обчислення вектор-функції правої частини системи. Тому у більшості чисельних методів, у першу чергу, розпаралелюється обчислення компонент вектор-функції правої частини системи рівнянь на вибраній кількості обчислювальних елементів комп'ютера.

Щоб забезпечити рівномірне завантаження обчислювальних елементів комп'ютера, необхідно виконати автоматичний розподіл обчислення значень компонент вектор-функції на *p* блоків, де *p* – кількість процесів [5].

На рівні паралелізму обчислювальних вузлів у середовищі MPI з розподіленою пам'яттю на *p* процесах реалізуються підзадачі 1)-6) на кожному кроці інтегрування алгоритму, а підзадача 7) – по завершенні процесу інтегрування. Обмін між даними для підзадач 2), 3) та 4) відбувається за стандартами MPI.

Оскільки підзадачі 1)-6) включають в себе виконання однорідних операцій над великою кількістю даних, то реалізується другий рівень паралелізму – рівень паралелізму потоків з використанням середовища OpenMP. Тут кожна з підзадач 1)-6) розпаралелюється між деякою кількіс-тю потоків (threads) на вільних ядрах з використанням спільної пам'яті.

На третьому рівні паралелізму – паралелізмі обробки даних – проводиться автоматична векторизація циклів за рахунок використання векторних процесорних пристроїв (VPU).

Експериментальне дослідження розроблених паралельних алгоритмів зазначених методів проводилися на паралельному комп'ютері Інпарком_xp**,** розробленому в Інституті кібернетики ім. В.М. Глушкова НАН України спільно з ДП «Електронмаш». Цей комп'ютер має наступні характеристики:

- процесори: Intel Xeon Phi 7210 (64 ядра) з частотою 1.3 ГГц;
- об'єм швидкої пам'яті MCDRAM: 16 Гб;
- обсяг оперативної пам'яті: 192 Гб;
- об'єм SSD накопичувача: 240 Гб.

Для дослідження розробленого паралельного алгоритму методу Ньютона розв'язувалась СНР:

$$\sum_{j=1}^{n} x_j - 0.5(3n+1) + 2x_i^2 - \\ -2\left(1 + 2\frac{i}{n} + \left(\frac{i}{n}\right)^2\right) = 0, \qquad (12)$$

при заданому початковому наближенні $x^0 = 1 + \dfrac{i+1}{2n}$, в області $D = \{a_i \le x_i \le b_i\}$, $i = 0,1,2,\ldots,n-1$.

У таблиці 1 представлені часи розв'язування СНР (12) порядку n = 10 000 з використанням багаторівневого паралелізму, тобто *p* MPI процесів на верхньому рівні паралелізму, *p_treads* тредів для директив OpenMP на другому рівні паралелізму та автоматична векторизація циклів за рахунок використання векторних процесорних пристроїв на третьому рівні паралелізму. В останньому стовпчику таблиці наведено часи розв'язування цієї ж СНР з використанням лише *p* MPI процесів.

Таблиця 1. Часи розв'язування СНР у сек.

p_treads	8	16	32	*p* MPI
p				
1	732,88	383,67	253,51	7785,00
4	331,27	203,55	213,69	1984,17
8	203,98	213,88	242,34	1000,00
16	214,41	241,73		538,44

Отже, використання багаторівневого паралелізму призводить до суттєвого скорочення часу розв'язування СНР.

Коефіцієнт прискорення $S_p = T_1 / T_p$,

де T_1 – час розв’язування послідовним алгоритмом; T_p – часи розв’язування на *p* ядрах та *p_threads* тредах для СНР (12) порядку *n* = 10 000 представлені на рисунку 1.

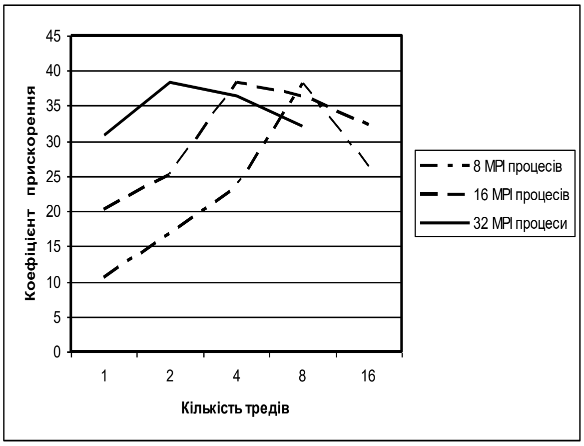


Рис. 1. Коефіцієнт прискорення

Результати, наведені на рис. 2, свідчать, що розроблений паралельний алгоритм забезпечує масштабованість, тобто час розв’язування задачі зменшується пропорційно росту кількості обчислювальних пристроїв. Найбільше прискорення порівняно з послідовним алгоритмом одержано при використанні *p* MPI процесів та *p_threads* тредів для директив OpenMP, для яких *p***p_threads* = 64, що дорівнює кількості ядер на 32 плитках (по 2 ядра на плитку). Тобто, для отримання максимального прискорення необхідно враховувати архітектуру комп’ютера.

Для дослідження розробленого паралельного алгоритму методу Рунге-Кутта 4-го порядку розв’язувалась задача Коші для системи звичайних диференціальних рівнянь:

$$\frac{du_i}{dt} = -\sum_{j=0}^{n-1} u_j - u_i + n(1+t) + 2 + t\,, \qquad (13)$$

з початковими умовами $u_i(0) = 1$ ($i = 0, 1, \ldots, n-1$), на інтервалі [0,0; 0,4].

Наведемо деякі з отриманих результатів.

Часи розв’язування СЗДР (13) порядку *n* = 19 968 з використанням багаторівневого паралелізму, тобто *p* MPI процесів на верхньому рівні паралелізму, *p_treads* тредів для директив OpenMP на другому рівні паралелізму та автоматичної векторизації циклів за рахунок використання векторних процесорних пристроїв на третьому рівні паралелізму представлені в таблиці 2. В останньому стовпчику таблиці наведено часи розв’язування цієї ж СЗДР з використанням лише *p* MPI процесів.

Таблиця 2. Часи розв’язування СЗДР у сек.

p_treads	8	16	32	*p* MPI
p				
1	83,93	45,53	24,01	617,26
2	66,70	34,29	19,61	306,30
4	28,17	15,69	15,80	153,22
8	15,39	14,11	15,09	76,91
16	13,24	14,39		37,05

Коефіцієнти прискорення для даної СЗДР порядку *n* = 19 968 представлено на рис. 2.

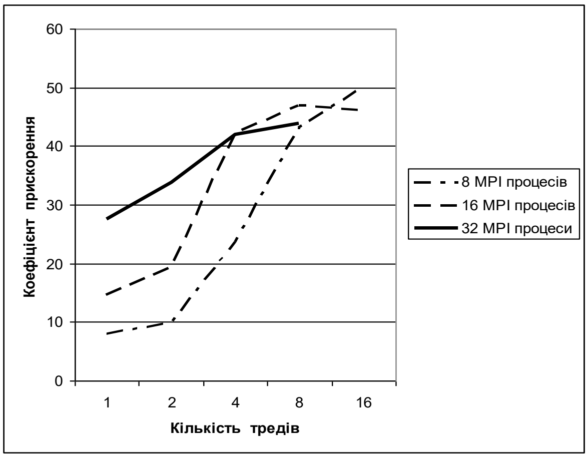


Рис. 2. Коефіцієнт прискорення

Результати, наведені на рис. 2, свідчать, що розроблений паралельний алгоритм забезпечує зростання коефіцієнта прискорення залежно від кількості процесів. Найбільше прискорення у порівнянні з послідовним алгоритмом одержано при використанні *p* MPI процесів та *p_threads* тредів для директив OpenMP, для яких *p***p_threads* = 128.

Висновки

Використовуючи багаторівневу модель паралельних обчислень з урахуванням особливостей архітектури комп’ютера, розроблено ефективні алгоритми та програми розв’язування СНР та задач з початковими умовами для СЗДР на паралельних комп’ютерах з процесорами Intel Xeon Phi другого покоління. При цьому час розв’язування задач суттєво скорочується, що дає можливість розв’язувати задачі високих порядків у реальному часі, які висуває сучасне життя перед наукою.

Література

1. Intel Xeon Phi Coprocessor System Software Developers Guide, revision 2.03. – 2012. – p. 201.
2. Intel Xeon Phi [Elektr. Resurs]. – Режим доступу: https://www.intel.com/content/www/us/en/products/processors/xeon-ph
3. Нестеренко А.Н., Химич А.Н., Яковлев М.Ф. (2006) Некоторые вопросы решения систем нелинейных уравнений на многопроцессорных вычислительных системах с распределенной памятью. *Вестник компьютерных и информационных технологий*, М.: 2006. – № 10. – С. 54-56.
4. Яковлев М.Ф., Герасимова Т.О., Нестеренко *А.Н.* (2009) Особливості розв’язування систем нелінійних та диференціальних рівнянь на паралельних комп’ютерах. *Питання оптимізації обчислень* (ПОО – XXXV). Праці міжнародного симпозіуму. – Київ: Інститут кібернетики ім. В.М. Глушкова НАН України, 2009. – Т.2.– С. 435-439.
5. Химич А.Н., Яковлев М.Ф., Герасимова Т.А. (2007) Некоторые вопросы решения систем обыкновенных дифференциальных уравнений на MIMD-компьютерах *Кибернетика и системный аналіз*. – 2007. – № 2. – С. 175-182.

References

1. Intel Xeon Phi Coprocessor System Software Developers Guide, revision 2.03. – 2012. – p. 201.
2. Intel Xeon Phi [Elektr. Resurs]. – Rezhym dostupu: https://www.intel.com/content/www/us/en/products/processors/xeon-ph
3. Nesterenko A.N., Himich A.N., Yakovlev M.F. (2006) Nekotoryie voprosyi resheniya sistem nelineynyih uravneniy na mnogoprotsessornyih vyichislitelnyih sistemah s raspredelennoy pamyatyu. Vestnik kompyuternyih i informatsionnyih tehnologiy, M.: 2006. – № 10. – S. 54-56.
4. Yakovlev M.F., Herasymova T.O., Nesterenko A.N. (2009) Osoblyvosti rozviazuvannia system neliniinykh ta dyferentsialnykh rivnian na paralelnykh kompiuterakh. Pytannia optymizatsii obchyslen (POO – XXXV). Pratsi mizhnarodnoho sympoziumu. – Kyiv: Instytut kibernetyky im. V.M. Hlushkova NAN Ukrainy, 2009. – T.2.– S. 435-439.
5. Himich A.N., Yakovlev M.F., Gerasimova T.A. (2007) Nekotoryie voprosyi resheniya sistem obyiknovennyih differentsialnyih uravneniy na MIMD-kompyuterah *Kibernetika i sistemnyiy analIz.* (2), 175-182.

RESUME

A.N. Nesterenko, T.O. Gerasimova, I.A. Baranov

Some approaches to develop parallel algorithms for solving tasks on computers with Intel Xeon Phi processors

Mathematical modeling is one of fundamental tools for the studying of various nature processes and phenomena arising in society, economy, science and engineering. Very often during the numerical modeling in various branches of science and engineering problems arise which are either non-linear systems (NLS) or initial-value problems for systems of ordinary differential equations (ODE).

The solving of such problems on computers requires an increase in computers productivity. The increase in computer performance is closely linked to an increase in the number of processors. However, an increase in amount of processors in parallel computers very often results both in the considerable growth of communicational losses and in the reduction in their productivity.

The requirements to high performance computing machinery surpass the potentialities of traditional parallel computers. Therefore, a special attention should be paid to the development of supercomputers possessing principally new architecture. Multi-core Intel Xeon Phi processors developed by Intel corporation belong to such class of supercomputers. The new generation processors Intel Xeon Phi x200 are processors-accelerators which could completely replace central processors in parallel computer.

The paper deals with some approaches to the development of parallel algorithms intended for the solving both of non-linear systems or initial-value problems for systems of ordinary differential equations on computers with multi-core Intel Xeon Phi x200 processors. These approaches are based on the employment of multi-level model of parallel computations which enables to considerably simplify the programming process.

Надійшла до редакції 01.10.2018